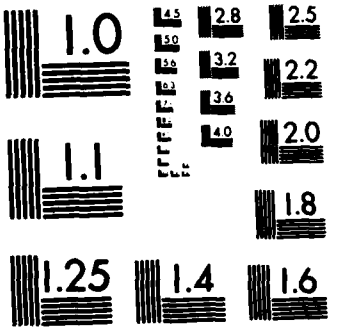
AD-A145 623 YEARLY REPORT FOR PROGRAMMING PRODUCTIVITY ENHANCEMENT 1/1
BY THE USE OF APPL.. (U) UNIVERSITY OF SOUTHERN
CALIFORNIA LOS ANGELES DEPT OF COMPUTE.. E HOROWITZ
UNCLASSIFIED AUG 84 AFOSR-TR-84-0813 AFOSR-82-0232 F/G 9/2 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A145 623

(9)

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS:
2a. SECURITY CLASSIFICATION AUTHORITY:	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:	
4. PERFORMING ORGANIZATION REPORT NUMBER(S):	5. MONITORING ORGANIZATION REPORT NUMBER(S): AFOSR-TR 84-0813
6a. NAME OF PERFORMING ORGANIZATION: University of Southern California; 6b. OFFICE SYMBOL (If applicable):	7a. NAME OF MONITORING ORGANIZATION: Air Force Office of Scientific Research
6c. ADDRESS (City, State and ZIP Code): Computer Science Department, University Park, Los Angeles CA 90089	7b. ADDRESS (City, State and ZIP Code): Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: AFOSR; 8b. OFFICE SYMBOL (If applicable): NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: AFOSR-82-0232
8c. ADDRESS (City, State and ZIP Code): Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A2	

11. TITLE (Include Security Classification): YEARLY REPORT FOR PROGRAMMING PRODUCTIVITY ENHANCEMENT BY THE USE OF APPLICATION /GENERATORS

12. PERSONAL AUTHOR(S): Ellis Horowitz

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Yr., Mo., Day)	15. PAGE COUNT
Interim	FROM 1/6/83 TO 31/5/84	AUG 84	4

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

During this period the investigator produced three papers with titles, "AdaRel --- A relational extension of Ada;" "The design of office information systems," and "High-level input-output facilities in database programming language."

DTIC FILE COPY

DTIC ELECTE SEP 19 1984 S D E

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE NUMBER (Include Area Code)	22c. OFFICE SYMBOL
Dr. Robert N. Buchal	(202) 767-4939	NM

DD FORM 1473, 83 APR — EDITION OF 1 JAN 73 IS OBSOLETE.

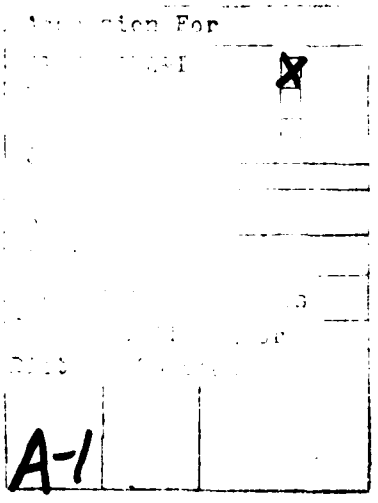
UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AFOSR-

Yearly Report For
Programming Productivity Enhancement by the Use of Application Generators
AFOSR-82-0232
Covering the Period:June 1, 1983 - May 31, 1984

Principal Investigator
Dr. Ellis Horowitz
Computer Science Department
University of Southern California
Los Angeles, California 90089
213-743-6453

submitted to
Dr. Robert Buchal
AFOSR/NP
Building 410
Bolling AFB, DC 20332
202-767-5025



Progress this Past Year

The past year has been a major one for this three year grant. The first year was spent largely doing preparatory work. During the second year we had the following list of accomplishements:

- we fully specified the data base extension to Ada, which we call AdaRel.
- we conceptualized the design of an office information system based upon forms.
- we began to investigate ways to provide within AdaRel the ability to create man-computer dialogues.

We started this past year by choosing Ada as our base language.The database language extensions we have designed are based upon the relational data model. The system is interfaced to a relational database management system via a new Ada type ***relation***. The language includes basic operations on relations, similar to those commonly available in database query languages, such as retrieval of data, updating of tuples and high-level operators to combine relations to form new ones. We have shown how Ada exception handling is naturally extended to allow integrity control of the relations. We also have devised language features that enable the sharing of objects among several users. The entire extension of Ada is reported on in our paper entitled *AdaRel: A Relational Extension of Ada*,[1].

The second major activity during this period has been the investigation of integrated office information systems. By studying the programming needs of offices, which are often data intensive, we hoped to be led towards new programming language facilities that support and enhance the database interface described above. From this study we concluded that there was a major area of software development which is inadequately supported by current programming languages. This is the area of software that interacts with users in real time. Programming "such "sessions" is becoming a standard activity and yet, conventional programming languages have operators that deal with characters and lines and not with screens or sequences of screens. A second point of inspiration that resulted from our study of office information systems is that the most common mechanism for the individual interacting with the software is the automated form. A "form" is something we all deal with often in our everyday affairs. An integrated office information system based upon forms offers a non-procedural way of programming office applications and so it is properly viewed as a manifestation of an application generator.

Pursuing the analogy, we have now defined the basic properties of a form and the operations that must be supported by a forms system if a system based upon it will be capable of describing the complete range of programming tasks. These include: 1. form template definition; 2. form template instantiation; 3. specifying actions on form instances such as mailing, copying, saving and triggering; 4. validation of forms; and 5. storage and retrieval of forms and their contents. This work has been summarized in our paper *The Design of Office Information Systems*,[2].

Our final work has focused on the question of how one can increase the power of programming languages for building man-computer dialogues. What we mean by such dialogues in general is that there is a network of screens of information that might potentially be communicated to the end-user of the program. Depending upon the end-users response, different screens are shown. The data on the screen, will often contain information from the database. Thus the following research questions were raised:

- how can one easily specify such networks of screens?
- how can one describe the layout of data of any particular screen?
- how can one provide editing capabilities for the input of responses from the screen?
- how is one to specify validity checking of these inputs?
- how can the program provide scrolling of the screen so that a great volume of data will be visible?

We have made a start on this latter issue and intend to continue working on it in the coming year. Our work is summarized in *High-Level Input/Output Facilities in a Database Programming Language*, [3].

Personnel

During this period the grant supported the Principal Investigator for 2 months during the summer and two graduate students: Mr. Alfons Kemper and Mr. Balaji Narasimhan. Mr. Kemper is near to completing his Ph.D. thesis and hence will be graduating late this summer.

References

[1] "AdaRel: A Relational Extension of Ada", Technical Report TR-83-218, Computer Science, USC, October, 1983, submitted to *ACM Trans. on Programming Languages*.

[2] "The Design of Office Information Systems", Technical Report TR-83-320, Computer Science, USC, October, 1983, to be submitted to *ACM Trans. on Office Systems*.

[3] "High-Level Input-Output Facilities in Database Programming Language", Computer Science Technical Report TR-84-307, USC, June 1984, submitted to *ACM Trans. on Office Systems*.